

Smart Piggy Bank

Introducere

Proiectul constă într-o pușculiță inteligentă controlată de un microcontroller ATmega. Sistemul permite accesul securizat prin introducerea unei parole de la un keypad 3x4, afișează mesaje pe un LCD1602 și controlează mecanismul de închidere cu ajutorul unui servomotor. Monedele introduse sunt sortate mecanic în funcție de dimensiune și detectate cu ajutorul unor senzori IR. Sistemul oferă feedback vizual prin LED RGB și feedback sonor prin buzzer. Scopul proiectului este să creez o pușculiță mai practică și mai interactivă decât una obișnuită. Aceasta va putea fi deschisă doar cu parolă, va sorta monedele automat după dimensiune și va oferi feedback utilizatorului prin LCD, LED RGB și buzzer.

Descriere generală

Sistemul este alcătuit din următoarele module:

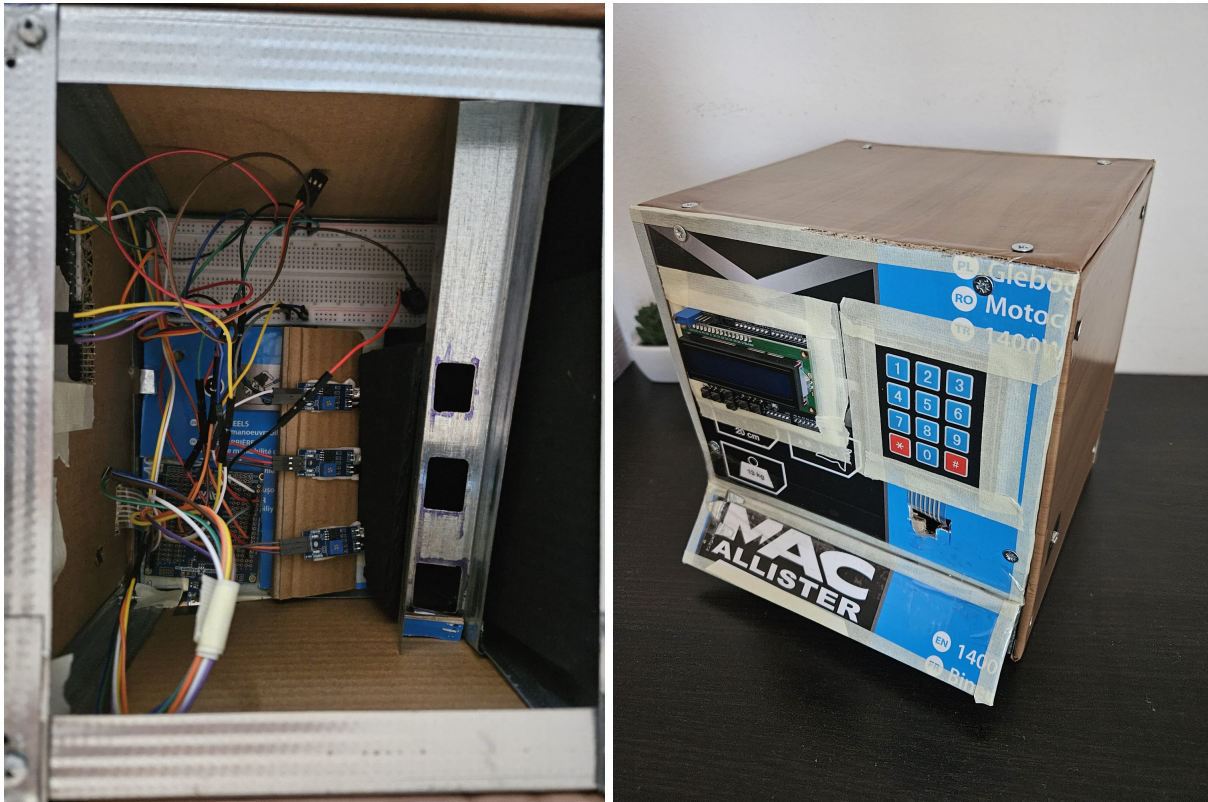
- **Modul de control** - placă ATmega
- **Modul de autentificare** - keypad 3x4
- **Modul de afișare** - LCD1602
- **Modul de închidere** - servomotor
- **Modul de detecție monede** - senzori IR
- **Modul de feedback** - buzzer

Fluxul principal al aplicației:

- utilizatorul introduce parola pe keypad
- microcontrollerul verifică parola
- LCD-ul afișează starea sistemului
- dacă parola este corectă, servomotorul deblochează mecanismul
- monedele sunt sortate mecanic și detectate cu senzori IR
- sistemul oferă feedback prin buzzer



Hardware Design



Listă de piese

Componentă	Rol
Placă ATmega	Controlul sistemului
Keypad 3x4	Introducerea parolei
LCD1602 shield	Afișarea mesajelor
Servomotor SG90 180°	Controlul mecanismului de închidere
Senzori IR	Detectarea monedelor
Buzzer activ 5V	Feedback sonor
Rezistori 220Ω / 330Ω	Limitarea curentului pentru LED
Breadboard MB-102	Realizarea montajului
Fire Dupont	Conectarea componentelor

În cadrul proiectului a fost realizată integrarea mai multor componente hardware folosind placa de dezvoltare ATmega328P Xplained Mini. Până în acest moment au fost implementate și testate următoarele funcționalități:

- afișarea informațiilor pe display-ul LCD1602;
- introducerea unei parole folosind keypad-ul matricial;
- detectarea monedelor folosind senzori IR;
- generarea de feedback sonor folosind buzzerul activ;
- utilizarea Timer2 pentru gestionarea operațiilor dependente de timp;
- actualizarea și afișarea sumei totale introduse.

Conexiuni hardware



LCD1602

Funcție LCD	Pin microcontroller	Rol
RS	PB0	Selectează dacă se transmite o comandă sau un caracter către LCD
EN	PB1	Semnal de validare; LCD-ul citește datele când acest pin este activat
D4	PD4	Linie de date pentru modul 4-bit
D5	PD5	Linie de date pentru modul 4-bit
D6	PD6	Linie de date pentru modul 4-bit
D7	PD7	Linie de date pentru modul 4-bit
VDD	5V	Alimentarea LCD-ului
VSS	GND	Masă comună
RW	GND	LCD-ul este folosit doar pentru scriere

LCD-ul este utilizat în modul 4-bit pentru reducerea numărului de pini utilizați. Pini PB0 și PB1 au fost aleși pentru semnalele de control, iar pini PD4–PD7 pentru liniile de date deoarece sunt grupați pe același port, ceea ce simplifică implementarea software.

Modulul LCD1602 folosit are potențiometrul inclus pentru reglarea contrastului, ceea ce permite ajustarea vizibilității caracterelor afișate.

Keypad 3x4

Funcție keypad	Pin microcontroller	Rol
ROW1	PB2	Linie de scanare pentru primul rând
ROW2	PB3	Linie de scanare pentru al doilea rând
ROW3	PB4	Linie de scanare pentru al treilea rând
ROW4	PB5	Linie de scanare pentru al patrulea rând
COL1	PD0	Intrare pentru prima coloană
COL2	PD1	Intrare pentru a doua coloană
COL3	PD2	Intrare pentru a treia coloană

Tastatura matricială este scanată secvențial prin activarea liniilor și citirea coloanelor. Liniile au fost conectate pe portul B deoarece sunt folosite ca ieșiri, iar coloanele pe portul D deoarece sunt citite ca intrări cu rezistențe interne pull-up activate.

Senzori IR

Senzor	Pin microcontroller	Rol
--------	---------------------	-----

Senzor 5 bani	PC5	Detectează moneda de 5 bani
Senzor 10 bani	PC4	Detectează moneda de 10 bani
Senzor 50 bani	PC3	Detectează moneda de 50 bani

Senzorii IR detectează trecerea monedelor prin modificarea stării logice a pinilor de intrare. Pinii PC3-PC5 au fost aleși deoarece erau disponibili și permit citirea simplă a semnalelor digitale.

Buzzer activ

Componentă	Pin microcontroller	Rol
Buzzer	PC2	Generează feedback sonor

Buzzerul activ este controlat printr-un singur pin digital. Pinul PC2 a fost ales deoarece era disponibil și permite pornirea/oprirea buzzerului prin semnal HIGH/LOW. Fiind un buzzer activ, acesta nu necesită generarea unei frecvențe PWM, ci doar alimentarea printr-un semnal digital.

Mecanismul de sortare al monedelor

Sistemul utilizează un mecanism mecanic de sortare a monedelor în funcție de dimensiunea acestora. Monedele introduse sunt direcționate către compartimente separate dedicate valorilor de 5 bani, 10 bani și 50 bani.

Sortarea este realizată printr-un traseu mecanic proiectat astfel încât monedele să fie separate pe baza diametrului lor. Fiecare compartiment este monitorizat cu ajutorul unui senzor IR care detectează trecerea monedei și transmite informația către microcontroller.

Pentru a îmbunătăți acuratețea detecției, interiorul compartimentelor din zona senzorilor IR a fost vopsit în negru. Această soluție reduce reflexiile luminii infraroșii produse de suprafața peretelui din spate și previne detectările false.

Software Design

Partea software a proiectului este implementată modular, fiecare componentă principală având propriul fișier sursă. Am ales această structură pentru că proiectul folosește mai multe componente hardware diferite, iar separarea codului face programul mai ușor de înțeles, testat și modificat.

Funcționalitățile software implementate sunt:

- citirea tastaturii matriciale 3x4;
- validarea parolei introduse de utilizator;
- schimbarea parolei prin mentinerea apăsată a tastei * timp de aproximativ 3 secunde;
- salvarea parolei în EEPROM, astfel încât aceasta rămâne memorată și după resetarea plăcii;
- afișarea mesajelor și a sumei totale pe LCD1602;

- controlul servomotorului pentru deschiderea si inchiderea trapei;
- detectarea monedelor de 5, 10 si 50 bani folosind senzori IR;
- actualizarea automata a sumei totale;
- feedback sonor prin buzzer;
- utilizarea intreruperilor pentru detectarea mai sigura a monedelor.

Stadiul actual al implementarii software

In acest moment partea software a proiectului este complet functionala. Aplicatia permite introducerea unei parole folosind keypad-ul matricial, validarea acesteia si controlul unui servomotor care deschide sau inchide trapa pusculitei.

Sistemul detecteaza monedele de 5, 10 si 50 bani folosind senzori IR. La detectarea unei monede, suma totala este actualizata si afisata pe LCD, iar utilizatorul primeste feedback sonor prin buzzer. Pentru a evita detectiile multiple ale aceleiasi monede a fost implementat un mecanism de cooldown software.

Parola poate fi schimbata de utilizator prin mentinerea tastei * timp de aproximativ 3 secunde. Pentru schimbarea parolei este necesara introducerea parolei vechi, dupa care utilizatorul poate introduce o parola noua. Noua parola este salvata in memoria EEPROM a microcontrollerului, astfel incat aceasta ramane disponibila si dupa repornirea sistemului.

De asemenea, in timpul introducerii parolei, utilizatorul poate apasa tasta # pentru a anula operatia si a reveni la ecranul principal cu suma totala.

Biblioteci folosite

Am ales sa implementez majoritatea functionalitatilor folosind biblioteci proprii.

Biblioteca pentru LCD implementeaza comunicatia cu afisajul LCD1602 in modul 4-bit. Aceasta permite afisarea sumei totale si a mesajelor pentru utilizator.

Biblioteca pentru keypad realizeaza scanarea tastaturii matriciale 3x4. Aceasta este folosita pentru introducerea parolei, anularea operatiilor si schimbarea parolei.

Biblioteca pentru motor controleaza servomotorul folosit pentru deschiderea si inchiderea trapei. Controlul se face prin semnal PWM.

Biblioteca pentru buzzer gestioneaza feedback-ul sonor. Buzzerul este activat diferit in functie de moneda detectata.

Biblioteca pentru timer este folosita pentru masurarea timpului in milisecunde. Aceasta este necesara pentru cooldown-ul detectiei monedelor, pentru gestionarea buzzerului si pentru detectarea apasarii lungi a tastei *.

Biblioteca utils contine functii ajutatoare pentru lucrul cu GPIO, precum configurarea unui pin ca input/output, activarea pull-up-ului intern, setarea unui pin pe HIGH/LOW si citirea starii unui pin.

Am folosit si biblioteca avr/eeprom.h pentru salvarea parolei in memoria EEPROM a microcontrollerului.

Elementul de noutate al proiectului

Elementul principal de noutate il reprezinta combinarea unei pusculite clasice cu functii inteligente de securitate si monitorizare.

Proiectul nu doar colecteaza monede, ci le sorteaza mecanic in functie de dimensiune, le detecteaza individual si actualizeaza automat suma economisita. In plus, accesul la continutul pusculitei este protejat prin parola, iar aceasta poate fi modificata direct de utilizator fara a reprograma dispozitivul.

Salvarea parolei in EEPROM permite pastrarea configuratiei chiar si dupa intreruperea alimentarii.

Utilizarea functionalitatilor din laborator

In cadrul proiectului au fost utilizate mai multe concepte studiate la laborator.

GPIO a fost folosit pentru controlul LCD-ului, citirea tastaturii matriciale, controlul buzzerului si citirea senzorilor IR. De exemplu, pinii conectati la senzori sunt configurati ca intrari cu pull-up intern, iar pinul buzzerului este configurat ca iesire digitala.

Timerele hardware au fost utilizate pentru generarea semnalului PWM necesar servomotorului si pentru implementarea functiilor dependente de timp. Timerul este folosit pentru cooldown-ul detectiei monedelor, pentru semnalele buzzerului si pentru detectarea apasarii lungi a tastei *.

Intreruperile au fost folosite pentru detectarea monedelor. Initial detectia era realizata prin polling, dar unele impulsuri foarte scurte puteau fi ratate. Din acest motiv am folosit Pin Change Interrupt pe pinii senzorilor IR. Astfel, microcontrollerul poate reactiona imediat cand un senzor isi schimba starea.

USART a fost folosit in etapa de dezvoltare pentru depanare. Prin monitorul serial am afisat starea senzorilor si a variabilelor interne, ceea ce m-a ajutat sa observ ca unele impulsuri erau prea scurte si erau ratate prin polling.

EEPROM a fost folosita pentru stocarea permanenta a parolei. Astfel, parola schimbata de utilizator ramane salvata si dupa resetarea placii.

Scheletul proiectului si interactiunea dintre functionalitati

Aplicatia este organizata modular. Fiecare componenta hardware are propriul fisier sursa, ceea ce face codul mai usor de urmarit:

- lcd.c / lcd.h - afisarea mesajelor pe LCD1602;

- keypad.c / keypad.h - citirea tastaturii matriciale;
- motor.c / motor.h - controlul servomotorului;
- buzzer.c / buzzer.h - feedback sonor;
- timer.c / timer.h - masurarea timpului;
- uart.c / uart.h - comunicatie seriala pentru debug;
- utils.c / utils.h - functii generale pentru GPIO;
- main.c - logica principala a aplicatiei.

La pornire, sistemul initializeaza toate modulele hardware, incarca parola din EEPROM si afiseaza suma totala pe LCD. In bucla principala, programul verifica apasarile de taste, proceseaza evenimentele generate de senzori si actualizeaza buzzerul.

Cand o moneda trece prin dreptul unui senzor IR, este generata o intrerupere. Intreruperea nu actualizeaza direct suma, ci seteaza un flag. Acest flag este apoi procesat in bucla principala, unde se verifica daca trapa este inchisa si daca a trecut timpul de cooldown. Daca aceste conditii sunt indeplinite, suma se actualizeaza si se porneste buzzerul.

Pentru deschiderea trapei, utilizatorul apasa tasta #. Daca trapa este inchisa, sistemul cere parola. Daca parola este corecta, servomotorul deschide trapa. Daca trapa este deja deschisa, o noua apasare pe # o inchide fara a mai cere parola.

Pentru schimbarea parolei, utilizatorul tine apasata tasta * timp de aproximativ 3 secunde. Sistemul cere parola veche, apoi parola noua, iar noua valoare este salvata in EEPROM.

Validarea functionalitatilor

Functionalitatile au fost validate incremental. Mai intai am testat separat afisajul LCD, keypad-ul, buzzerul, servomotorul si senzorii IR. Dupa ce fiecare componenta a functionat individual, am integrat modulele in aplicatia principala.

Pentru detectia monedelor am folosit initial afisari prin USART in monitorul serial. Astfel am putut vedea valorile citite de pe pini si am observat ca unele impulsuri ale senzorilor erau prea scurte pentru a fi detectate sigur prin polling. Dupa trecerea la Pin Change Interrupt, detectia a devenit mai stabila.

Am testat si scenariile principale de utilizare:

- introducerea monedelor si actualizarea sumei;
- detectia separata pentru 5 bani, 10 bani si 50 bani;
- deschiderea trapei cu parola corecta;
- anulara introducerii parolei folosind tasta #;
- inchiderea trapei fara parola;
- schimbarea parolei;
- resetarea placii si verificarea faptului ca parola noua ramane salvata.

Calibrarea senzorilor

Calibrarea senzorilor IR a fost realizata experimental. Am introdus de mai multe ori monedele prin

traseul mecanic si am verificat daca senzorii detecteaza corect trecerea acestora.

O problema observata a fost aparitia reflexiilor in interiorul compartimentelor. Pentru a reduce detectiile false, zona din spatele senzorilor IR a fost vopsita in negru. Aceasta modificare a redus reflexiile luminii infrarosii si a facut detectia mai stabila.

In software, am adaugat si un cooldown dupa fiecare detectie. Acesta impiedica numararea aceleiasi monede de doua ori in cazul in care senzorul genereaza mai multe tranzitii rapide.

Optimizari realizate

O prima optimizare a fost trecerea de la detectia prin polling la detectia bazata pe intreruperi. Prin polling, programul verifica periodic starea senzorilor, dar exista riscul ca impulsurile scurte sa fie ratate. Prin folosirea Pin Change Interrupt, schimbarea starii senzorului este capturata imediat.

O alta optimizare a fost gestionarea buzzerului fara delay-uri blocante. In loc sa opresc complet programul cat timp buzzerul suna, am folosit o logica bazata pe timp. Astfel, programul poate continua sa citeasca tastatura si sa proceseze senzorii.

Codul a fost impartit in module separate pentru a fi mai usor de inteles si de modificat. Aceasta organizare ajuta si la depanare, deoarece fiecare componenta poate fi testata separat.

Utilizarea laboratoarelor in proiect

GPIO

GPIO este folosit pentru configurarea pinilor digitali. Pentru a face codul mai usor de citit, am creat functii ajutatoare in utils.c.

Exemplu de functie pentru configurarea unui pin ca intrare cu pull-up intern:

```
void pin_input_pullup(volatile uint8_t *ddr,
                     volatile uint8_t *port,
                     uint8_t pin) {
    *ddr &= ~(1 << pin);
    *port |= (1 << pin);
}
```

Aceasta functie modifica registrul DDR pentru a seta pinul ca intrare si registrul PORT pentru a activa rezistenta interna de pull-up.

Functia este folosita pentru initializarea senzorilor IR:

```
void sensors_init(void) {
    pin_input_pullup(&DDRC, &PORTC, SENSOR_5BANI);
    pin_input_pullup(&DDRC, &PORTC, SENSOR_10BANI);
    pin_input_pullup(&DDRC, &PORTC, SENSOR_50BANI);
}
```

Astfel, senzorii sunt conectati pe pinii PC3-PC5 si sunt cititi ca intrari digitale.

Timere

Timerul este folosit pentru masurarea timpului in milisecunde:

```
ISR(TIMER1_COMPA_vect) {
    millis_counter++;
}
```

Aceasta valoare este folosita apoi pentru cooldown, buzzer si apasarea lunga a tastei *.

Intreruperi

Pentru detectia monedelor am folosit Pin Change Interrupt, deoarece impulsurile senzorilor pot fi foarte scurte:

```
ISR(PCINT1_vect) {
    uint8_t current = PINC;
    uint8_t changed = current ^ last_pin_c;

    if ((changed & (1 << SENSOR_5BANI)) &&
        !(current & (1 << SENSOR_5BANI))) {
        coin5_flag = 1;
    }

    last_pin_c = current;
}
```

In intrerupere nu actualizez direct suma, ci doar setez un flag. Logica mai mare este tratata in bucla principala.

Detectia monedelor

Dupa ce intreruperea seteaza flag-ul, moneda este procesata in bucla principala:

```
if (coin5_flag) {
```

```
coin5_flag = 0;

if (trapa_deschisa == 0 &&
    now - last_detect_5 >= COIN_COOLDOWN) {
    last_detect_5 = now;
    total_bani += 5;
    lcd_print_money(total_bani);
    buzzer_start_sound(1);
}
}
```

Cooldown-ul previne numararea aceleiasi monede de doua ori.

Controlul servomotorului

Servomotorul este controlat prin PWM. Unghiul este transformat intr-o valoare pentru registrul OCR2B:

```
void servo_angle(uint8_t angle) {
    OCR2B = 8 + ((uint16_t)angle * 30) / 180;
}
```

In proiect folosesc `servo_angle(0)` pentru trapa inchisa si `servo_angle(180)` pentru trapa deschisa.

Salvarea parolei in EEPROM

Parola este salvata in EEPROM pentru a ramane memorata dupa resetarea placii:

```
void password_save(void) {
    eeprom_update_block(password, ee_password, PASSWORD_LEN + 1);
    eeprom_update_byte(&ee_password_valid, 0xA5);
}
```

La pornire, parola este incarcata din EEPROM. Daca nu exista o parola valida, se foloseste parola initiala `3456`.

Schimbarea parolei

Pentru schimbarea parolei, utilizatorul introduce parola veche, apoi o parola noua:

```
if (strcmp(old_pass, password) != 0) {
    lcd_print("Parola gresita");
    return;
}
```

```
}  
  
strcpy(password, new_pass);  
password_save();
```

Astfel, parola noua este salvata permanent.

Buzzer

Buzzerul este controlat fara delay-uri lungi, folosind timpul curent:

```
if (now - beep_last_time >= BEEP_ON_TIME) {  
    buzzer_off();  
    beep_state = 0;  
    beep_done++;  
    beep_last_time = now;  
}
```

Am ales aceasta metoda pentru ca programul sa poata continua sa citeasca tastatura si senzorii in timp ce buzzerul suna.

Keypad

Keypad-ul este folosit pentru introducerea parolei si pentru comenzile speciale:

```
char key = keypad_get_key_once();  
  
if (key == '#') {  
    // cere parola sau inchide trapa  
}
```

Tasta `#` este folosita pentru deschiderea/inchiderea trapei, iar tasta `*` tinuta apasat porneste schimbarea parolei.

Rezultate Obținute

Proiectul final permite colectarea și monitorizarea monedelor de 5, 10 și 50 bani. Monedele sunt sortate mecanic și detectate cu ajutorul senzorilor IR, iar suma totală este afișată în timp real pe LCD.

Accesul la conținutul pușculiței este protejat prin parolă. Trapa poate fi deschisă doar după introducerea parolei corecte, iar parola poate fi schimbată direct de utilizator folosind keypad-ul. Noua parolă este salvată în memoria EEPROM și este păstrată chiar și după repornirea sistemului.

În urma testelor efectuate, detectarea monedelor s-a dovedit stabilă prin utilizarea întreruperilor hardware și a mecanismului de cooldown software. Toate funcționalitățile principale au fost testate cu succes pe montajul final.

Concluzii

Prin realizarea acestui proiect am aprofundat utilizarea perifericelor microcontrollerului ATmega328P, precum GPIO, timere, PWM, întreruperi și EEPROM.

Una dintre principalele provocări a fost detectarea fiabilă a monedelor. Inițial au existat situații în care unele impulsuri generate de senzori erau ratate, însă problema a fost rezolvată prin utilizarea Pin Change Interrupt și a unui mecanism de cooldown software.

Proiectul final îndeplinește obiectivele propuse inițial și demonstrează integrarea mai multor componente hardware într-un sistem funcțional și interactiv.

Download

Repository GitHub: <https://github.com/Eugen2613/Smart-Piggy-Bank>

Jurnal

Data	Activitate
27.04.2026	Alegerea temei proiectului
05.05.2026	Crearea paginii wiki
07.05.2026 - 10.05.2026	Implementarea hardware - conexiuni si realizarea carcasei
19.05.2026 - 21.05.2026	Implementarea software: detectia monedelor cu intreruperi, controlul servomotorului, gestionarea parolei, schimbarea parolei si salvarea acesteia in EEPROM

Bibliografie/Resurse

Listă cu documente, datasheet-uri, resurse Internet folosite, eventual grupate pe **Resurse Software** și **Resurse Hardware**.

[Export to PDF](#)

From:

<http://ocw.cs.pub.ro/courses/> - **CS Open CourseWare**

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2026/alexandru.jipa2803/eugen.cantaragiu>



Last update: **2026/05/21 15:22**